

MEMORY CONSTRAINED ANT COLONY SYSTEM FOR TASK SCHEDULING IN GRID COMPUTING

¹T.Kokilavani, ² Dr. D.I. George Amalarethinam

¹ Research Scholar, Bharathiar University, Coimbatore, India

vani78_ram@yahoo.com

²Associate Professor & Director, Department of MCA, Jamal Mohamed College, India

di_george@jmc.edu

ABSTRACT

Grid computing solves the increasing need of scientific, engineering and research problems. It combines the geographically distributed resources to solve a computation intensive problem which cannot be solved using a single resource. Resource sharing requires more optimized algorithmic structure, otherwise the response time is increased and the resource utilization is reduced. In order to avoid such reduction in the performance of the grid system, an optimal resource sharing algorithm is required. The ACO solves many engineering problems and provides optimal result which includes Travelling Salesman Problem, Network Routing, and Scheduling. This paper proposes Load Shared Ant Colony Optimization (LSACO) which shares the load among the available resources. The proposed method considers memory requirement as a parameter to distribute the load among Grid resources. LSACO reduces the overall response time and increases the resource utilization and number of tasks scheduled. The proposed method has been tested for different types of tasks and resources.

KEYWORDS

Grid Computing, Ant Colony Optimization, Task Scheduling, Resource Sharing

1. INTRODUCTION

As the scientific problem grows very complex in the modern computing technology, it requires more computing power and more storage space. Based on these basic requirements, an organization requires higher computational resource when dealing with current technological methodology. The past technologies such as distributed computing, parallel computing are not suitable for recent advancement. Therefore, the Grid computing is proposed as effective resource management to the organization.

In grid computing, the network status and the resource status are to be managed effectively. If the network status or resource status are not in feasible level, then the total computation time will be increased dramatically. In grid computing, the user will encounter thousands of computers to utilize in effective and efficient manner. The Grid architectures are serving as a middleware technology for various purposes like resource allocation management, job scheduling, data management, security and authorization. Programming in the grid computing involves more complexities which not only require a single-machine application but needs additional features.

Grid resource brokers [1] are responsible for gathering resource information, discovery, resource management and job scheduling. Various scheduling methods exist which try to optimize time,

cost, resource utilization and load balancing. Figure 1 shows the research activities related to Grid scheduling methods.

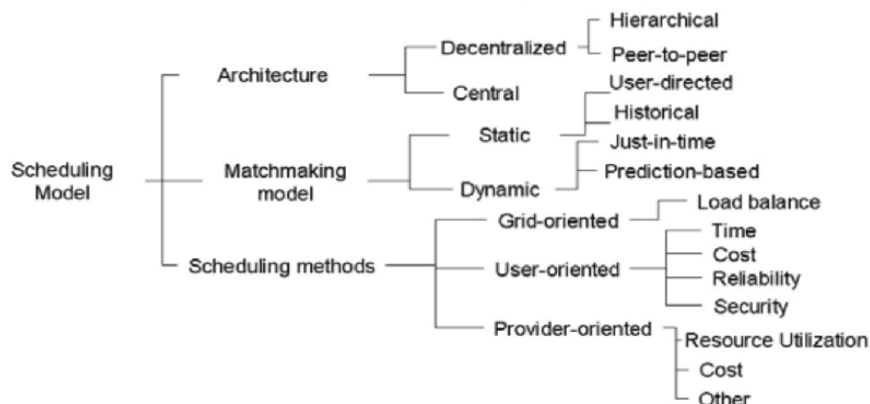


Figure 1. Various Research Activities in the Grid Environment

The purpose of task scheduling in the grid computing is to balance the load of entire grid system in such a way that completing all the assigned workload as soon as possible and feasible than other system. Job scheduling in the grid environment is a very important issue. The Non-traditional algorithms differ from the conventional traditional algorithms in that it produces optimal results in a short period of time [2]. There is no best scheduling algorithm for all grid computing systems. An alternative is to select an appropriate scheduling algorithm to use in a given grid environment based on the characteristics of the tasks, machines and network heterogeneity. The Grid scheduler should consider the QoS requirement of a task to find a perfect match of resource [3]. The QoS constraint is the responsibility of scheduler and it does not focus on the objective function. This paper proposes an efficient job scheduling algorithm based on Ant Colony Optimization which considers memory as a QoS requirement and shares the load among the resources for optimizing the resource usage in the grid environment.

2. RELATED WORK

Braun et al [4] have studied the relative performance of eleven heuristic algorithms for task scheduling in grid computing. They have also provided a simulation basis for researchers to test the algorithms. The simple algorithms proposed by Braun are Opportunistic Load Balancing (OLB), Minimum Execution Time (MET), Minimum Completion Time (MCT), Min-Min, Max-min and Genetic Algorithm.

Wei-Neng et al [5] proposed an ACO based task scheduling which considers the scheduling of tasks in terms of more than one quality of service (QoS) parameters. The ACO proposed by them enables the users to specify their QoS preferences as well as define the minimum QoS thresholds for a certain applications.

Ruay-Shiung Chang et al [6] proposed a Balanced Ant Colony Optimization (BACO) algorithm for job scheduling in the Grid environment while trying to minimize the makespan of a given set of jobs. According to the author the experimental results shows that BACO can outperform other job scheduling algorithms.

AliEn RB [7] is a Grid Broker which handles File transfer optimization, fault tolerance by multithreading, and Push and pull task assignment. In Apples [1], the Parameter study support, event-driven rescheduling, Centralized adaptive scheduling with heuristics, and self-scheduled work queues are handled. In EZ-GRID Broker [8], job handling, transparent file transfer, self-

information service with dynamic and historical data, Policy Engine Framework for provider policies are proposed. In GRID BUS Grid Service system [14], Failure management and application recovery, parameter study, API support, Economy-based and data aware scheduling are focused for solving. The GRUBER [15] handles SLA-based resource sharing in multi-VO environment, disk quota considerations, internal site monitoring feature, and various user oriented policies.

A load balancing algorithm aims to increase the utilization of resources with light load or idle resources thereby freeing the resources with heavy load [7]. The algorithm tries to distribute the load among all the available resources. At the same time, it aims to minimize the makespan with the effective utilization of resources. Only few of scheduling algorithms for grid task are focusing the problems with a variety of QoS parameter.

Stefka et al [10] proposed a high throughput computing scheduling algorithm based on ACO. The scheduling algorithm designed by them is for distributed systems shared asynchronously by both remote and local users.

Graham Ritchie et al [11] proposed an ant colony optimization (ACO) algorithm that, when combined with local and tabu search, can find shorter schedules on benchmark problems than other techniques found in the literature. From the results they show that the ACO approach can consistently find shorter makespans than any of the other approaches for all classes of ETC matrix tested.

Zhihong Xu et al [13] suggested that the structure of Grid makes the QoS of Grid more difficult than network and distribute computing environment. They proposed a category scheduling based on ant algorithm that makes use of user category, resource category and task category. The algorithm proposed by them improved the successful scheduling rate, system performance and Quality of Service (QoS) to a certain extent.

3. ANT COLONY SYSTEM

The ant algorithm is inspired from the behaviour of real ants. A colony of cooperative ants moves from their nest to find food. The ants deposit a chemical pheromone on its path while moving from nest to food. When more ants move in the same direction the strength of pheromone increases on the path. Other ants use this pheromone value to choose the shortest path. The idea of ants was used by Marco Dorigo to develop an Ant Colony System which aimed to search for an optimal path in a graph [13]. He formed artificial ants to find a solution for the Travelling salesman problem. The algorithm used the past history to update the pheromone value. Based on the pheromone value the shortest path was identified. The artificial ants use the state transition rule to select a resource in grid computing. The state transition rule used by ant system, called a random-proportional rule, gives the probability with which ant k chooses the resource r for job s . Pheromone value of a resource is updated if that particular resource is selected for scheduling.

J.L. Deneubourg et al proposed an ant colony system based on Argentine ants. According to the author most of the ants lay a pheromone trail only when they return from food source to nest. But argentine ants lay the pheromone trail both when leaving from and returning to the nest from food source. They proposed a model based on the behavior of these ants and formulated an equation by assuming that two paths are available for an ant. Figure 2 shows the two bridges and the formula is given in Equations (1) and (2).

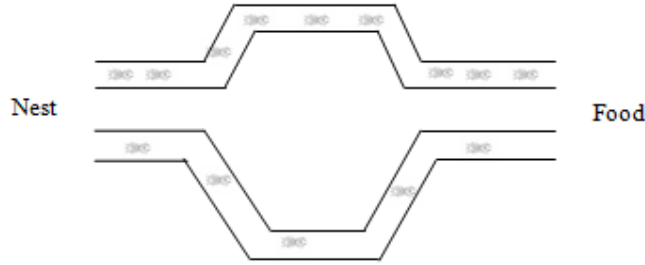


Figure 2. Ant moving from Nest to Food

$$Prob_A = \frac{(k + A_i)^n}{(k + A_i)^n + (k + B_i)^n} \quad (Prob_A + Prob_B = 1) \quad Eq(1)$$

$$A_{i+1} = A_i + \delta, \quad B_{i+1} = B_i + (1 - \delta) \quad (A_i + B_i = i) \quad Eq(2)$$

Where δ is a stochastic variable that takes a value of 1 or 0 based on the probability of $Prob_A$ and $Prob_B$ respectively. A is increased by 1 unit if ant chooses path A, otherwise B is increased by 1 unit if ant chooses path B. Based on the values on n and k , successive ants choosing the path may vary.

4. LOAD SHARED ANT COLONY OPTIMIZATION

This paper proposes a new load sharing approach based on the behavior of argentine ants. In deneubourg work they propose that the ants choose either path A or path B based on the probability value. This work considers path A as resource1 and path B as resource2. It also assumes that both resources are capable of executing the given tasks. The resources should be chosen based on the RAM requirement of task and the availability of RAM in the given resource. The traditional method allots any one resource based on the task requirements. In Ant Colony Optimization also, based on the probability and the requirements the task choose either resource1 or resource2. But this paper aims to share the load on both resources which will minimize both the makespan and wait time of the tasks. So it subdivides the given task and allots a percentage of tasks to both resources based on the probability value obtained by applying the random proportional rule. The random proportional rule used to identify the percentage of task allotted in a resource is given in Equations (3) and (4).

$$P_1 = \frac{(R_1 + k)^h}{(R_1 + k)^h + (R_2 + k)^h} \quad Eq(3)$$

$$A_1 = P_1 * TR_i \quad Eq(4)$$

Where A_1 is the amount of task allotted in resource1 and TR_i is the memory requirement of task_i. Based on the value of the coefficients k and h the probability of choosing a resource for scheduling a task will change. The value of k and h should be carefully chosen so that it shares the load among the resources. A deep study was done by varying the values of k and h and it was identified that by choosing the values of $k=1$ and $h=0$ load balancing is achieved.

The formula can be generalized if more number of resources is available. The pheromone value of each resource is updated based on the percentage of task allotted. The higher percentage the resource allotted, the value of pheromone incremented more. The probability rule used for more number of resources is given in Equation 5.

$$P_j = \frac{(R_i + k)^h}{\sum_{i=1}^n (R_i + k)^h} \quad \text{Eq (5)}$$

The tasks scheduled here are considered to be Meta tasks. Meta tasks don't have dependency relationship among them. So they can be easily subdivided and shared on many resources for parallel execution. The algorithm for the proposed work is as follows:

Procedure LSACO

1. Initialize ACO Parameters
2. For each resource
3. Read the Resource characteristics (available memory)
4. Initialize the pheromone value
5. End for
6. For each task
7. Read task requirement TR_i
8. For each resource calculate the probability value

$$P_j = \frac{(R_i + k)^h}{\sum_{i=1}^n (R_i + k)^h}$$

9. Calculate allotment percentage based on task requirement using A_j

$$A_j = P_j * TR_i$$

10. Update pheromone value for each resource if it is allotted

$$R_i = R_i + 1$$

11. End for
12. End for

5. EXPERIMENTS AND RESULTS

The proposed work is implemented in Java and it is shown in figure 3. A simulation tool has been developed to generate various numbers of heterogeneous tasks and resources. The heterogeneous task list, its requirement, heterogeneous resource list and its characteristic obtained from the simulation tool is taken by the LSACO algorithm and scheduling is produced. Traditional scheduling algorithm is also executed for the same set of tasks and resources. Tests are done for different cases. The experiments show that the response time for LSACO is lesser than the traditional method in all test cases and it does not vary based on the number of jobs and resources or heterogeneity of jobs and resources. The traditional methods will not schedule the task if the task requirement is more than the available memory size of the large resource because they choose any one resource and not all resources for executing a task. But LSACO will look for the sum of available memory size in all available resources and it will reject the task for scheduling only if the task requirement is more than the available memory size of the sum of all resources.

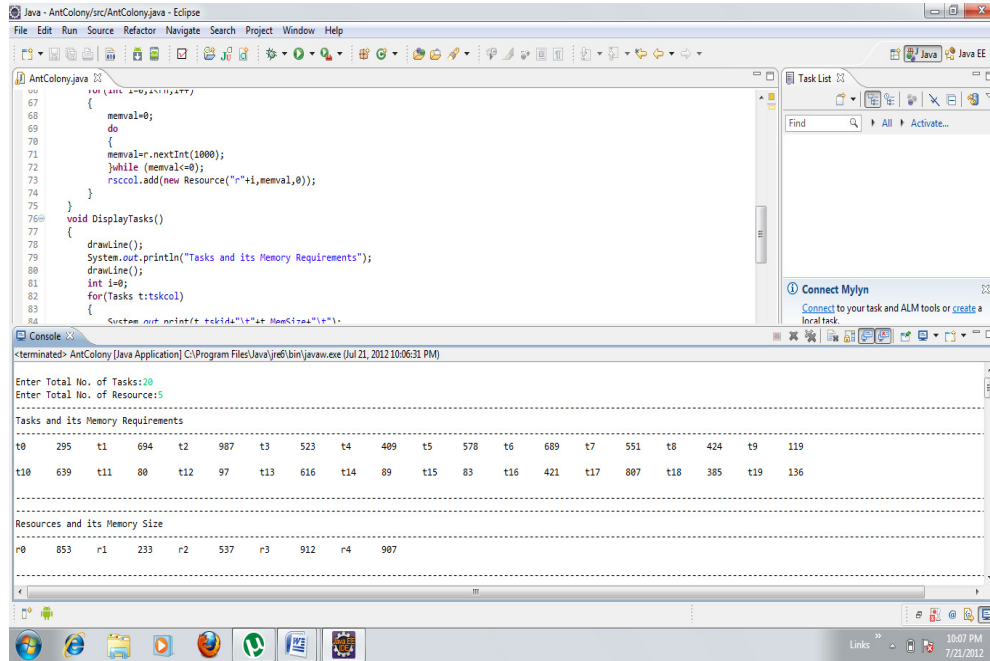


Figure 3. LSACO implementation

The test results are tabulated in table 1 and the improvement of LSACO over traditional method in response time, resource utilization and number of tasks scheduled is shown using the graphs in figures 4, 5, 6, 7, 8 and 9.

Table 1. Response Time and Resource Utilization

No. Of Tasks	No. Of Resources	Traditional Method			LSACO		
		Tasks Not Scheduled	Resources Used	Response Time	Tasks Not Scheduled	Resources Used	Response Time
10	5	T0, T4	R0	2189	-	R0 – R4	732.6
15		-	R0, R1	7687	-	R0 – R4	1537.4
20		T3, T8, T12	R0, R3, R4	6702	-	R0 – R4	1859.3
15	2	-	R1	6576	-	R0 , R1	5788.6
	4	T4	R0	5320	-	R0 – R3	1570.75
	6	T1, T3, T5, T7	R0, R5	4870	-	R0 – R5	1387.7

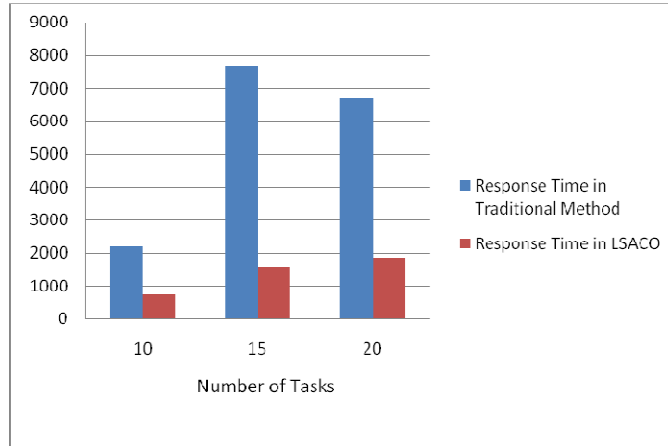


Figure 4. Graph Showing Response Time For Scalable Tasks

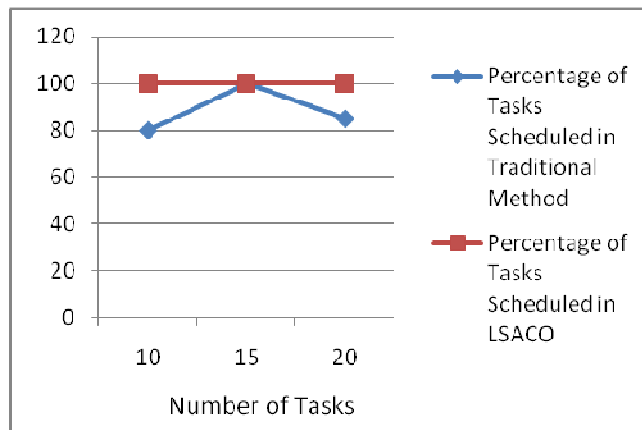


Figure 5. Task Schedule Chart For Scalable Tasks

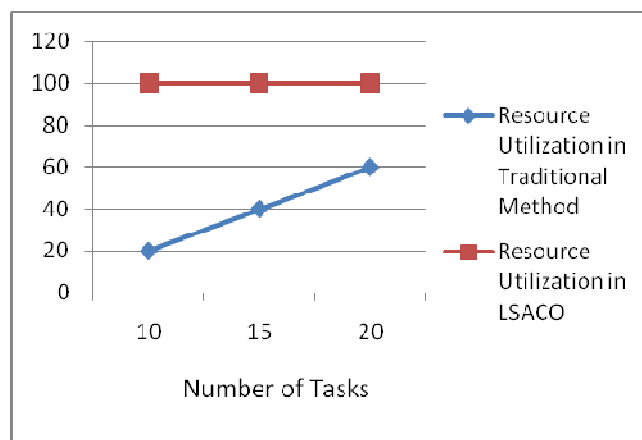


Figure 6. Resource Utilization Chart For Scalable Tasks

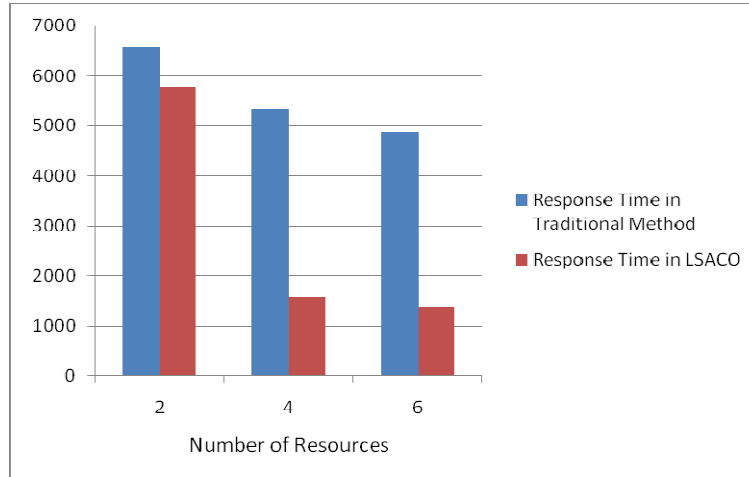


Figure 7. Graph Showing Response Time For Scalable Resources

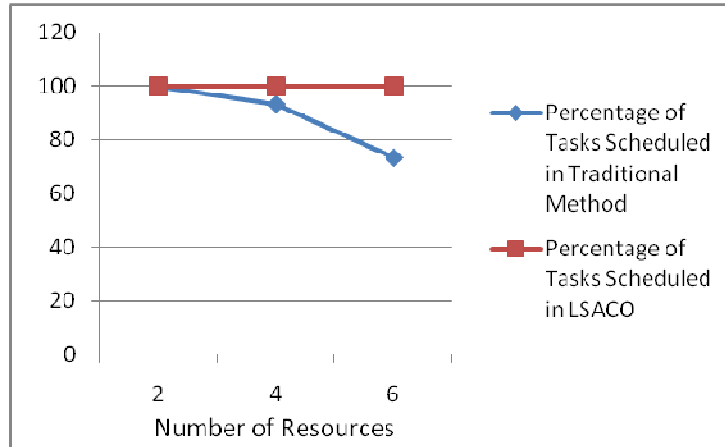


Figure 8. Task Schedule Chart For Scalable Resources

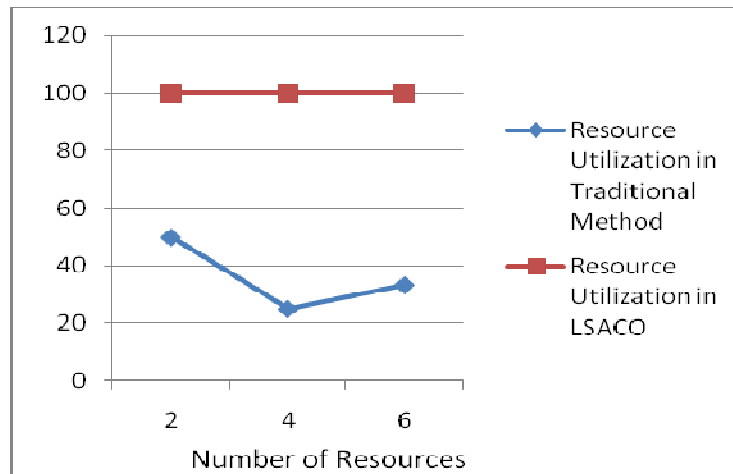


Figure 9. Resource Utilization Chart For Scalable Resources

6. CONCLUSION

The Ant Colony Optimization has proved to produce optimized results for many problems in science and engineering area. In this paper a new approach based on the argentine ant's behavior is proposed. The traditional methods try to reduce the overall response time by giving an optimized schedule. But they fail to produce a load balanced schedule. Moreover they do not share the load among the available resources. Thus the proposed method uses Ant Colony System approach to share the load among the resources for Meta Tasks. The proposed work uses one of the QoS parameter memory requirements to subdivide the task and schedule them in parallel among the available Grid resources. The experiments and results show that LSACO reduces response time of tasks. This method also shows that it increases the number of tasks scheduled and number of resources utilized. That is the idle time of resources is minimized. The work can be further extended by considering ETC matrix as another parameter to get the perfect wait time and response time of the tasks. It can also be extended for dependent tasks using DAG approach.

REFERENCES

- [1] H. Casanova, G. Obertelli, F. Berman, R. Wolski, "The AppLeS parameter sweep template: user-level middleware for the grid", Proceedings of the ACM/IEEE Conference on Supercomputing, 2003
- [2] Kokilavani.T and George Amalarethinam.D.I, "Applying Non-Traditional Optimization Techniques to Task Scheduling in Grid Computing", International Journal of Research and Reviews in Computer Science, Vol. 1, No. 4, Dec 2010, pp. 34 – 38
- [3] Amit Agarwal and Padam Kumar, "Multidimensional Qos Oriented Task Scheduling In Grid Environments", International Journal of Grid Computing & Applications (IJGCA) Vol.2, No.1, March 2011, pp 28 - 37
- [4] Braun, T.D., Siegel, H.J., Beck, N., Boloni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., et al. "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems", Journal of Parallel and Distributed Computing, Vol. 61, No. 6, pp.810–837, 2001
- [5] Wei-Neng Chen, Student Member, IEEE, and Jun Zhang, Senior Member, IEEE, "An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 39, No. 1, January 2009, pp 29 – 43
- [6] Ruay-Shiung Chang, Jih-Sheng Chang, Po-Sheng Lin, "An ant algorithm for balanced job scheduling in grids", Future Generation Computer Systems, Vol. 25, pp.20–27, 2009
- [7] P. Saiz, A. Buncic, J. Peters, "AliEn Resource Brokers", Proceedings of the Third international Workshop on in High-Energy and Nuclear Physics (CHEP 03), 2003
- [8] Attila Kertész, Péter Kacsuk, "A Taxonomy of Grid Resource Brokers", pp 201 -210.
- [9] Kokilavani.T and George Amalarethinam.D.I, "Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing", International Journal of Computer Applications (0975 – 8887) Volume 20– No.2, April 2011
- [10] Stefka Fidanova and Mariya Durchova, "Ant Algorithm for Grid Scheduling Problem", Large scale computing, Lecture Notes in Computer Science Springer, No.3743, pp. 405 – 412, 2006.
- [11] Graham Ritchie and John Levine, "A hybrid ant algorithm for scheduling independent jobs in heterogeneous computing environments", 2004, American Association for Artificial Intelligence.

- [12] Zhihong Xu, Junhua Gu, “Research on Ant Algorithm Based Task Category Scheduling in Grid Computing”, Second International Conference on Intelligent Networks and Intelligent Systems, pp. 498 – 501, 2009
- [13] Marco Dorigo, Luca Maria Gambardella, “Ant Colony system: A Cooperative Learning Approach to the Travelling Salesman Problem”, IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, 1997, pp 1 – 24.
- [14] B. Sundaram and B. M. Chapman, “XML-Based Policy Engine Framework for Usage Grid Computing”, LNCS vol. 2536, Springer-Verlag, pp. 194-198, 2002
- [15] C. Dumitrescu, I. Foster, “GRUBER: A Grid Resource Usage SLA Broker”, 11th International Euro-Par Conference, LNCS, Springer, Vol. 3648, pp. 465-474, 2005

Authors

Dr. D. I. George Amalarethinam is working as Associate Professor and Director (MCA) in the Department of Computer Science, Jamal Mohamed College, Tiruchirappalli, India. He has 25 years of experience in teaching and 16 years of experience in research. He has published more than 50 research articles in the international/national conferences and journals. He has also presented a research article in Nepal. He has chaired many technical sessions and delivered invited talks in national/international conferences. He is an editorial board member in leading international journals. His interests include Parallel Computing, Grid Computing, Distributed Databases and Ad hoc Networks.



T. Kokilavani obtained her Bachelors Degree and Masters Degree in Computer Science from Bharathidasan University, Tiruchirappalli. She is currently doing research in the field of Grid Computing. She has presented papers in National conferences and International conferences. She has also published papers in International journals. Her interests include Parallel Computing, Grid Computing, Mobile Computing and Cloud Computing.

